

Universidade Federal de Minas Gerais - UFMG

Jackson Medeiros de Paula Pereira

Trabalho Prático - Enduro

Belo Horizonte / MG

Dezembro de 2018

1. INTRODUÇÃO

O trabalho prático da disciplina de Algoritmos e Estrutura de Dados I foi a criação de um jogo de corrida, similar ao “Enduro” do Atari 2600, utilizando a biblioteca Allegro, apresentada aos alunos durante o curso.

Essa biblioteca permite a elaboração de jogos 2D com dificuldade relativamente fácil, tendo como vantagem a aplicação dos conteúdos aprendidos ao longo do curso, como a manipulação de ponteiros e estruturas.

O Enduro é um jogo de corrida, onde o jogador pode se mover apenas dentro dos limites da pista e tem que ultrapassar o máximo de oponentes sem colidir com nada no cenário.

O trabalho apresentado teve inspiração em jogos clássicos como Top Gear focando em uma jogabilidade mais simples e com uma mecânica de evolução bem simples, a medida que o tempo passa, mais oponentes vão aparecendo, incrementando a dificuldade do jogador.

2. IMPLEMENTAÇÃO

2.1. ESTRUTURAS

Color: Estrutura utilizada para a manipulação de cores com o padrão RGB.

Essa estrutura está localizada no header color.h e conta com diversos métodos para operações vetoriais com a estrutura, como por exemplo, soma, subtração, multiplicação e interpolação linear.

Enemy: Estrutura utilizada para a manipulação dos oponentes no jogo.

Estrutura utilizada para armazenar os dados dos oponentes do jogador.

2.2. MÉTODOS E FUNÇÕES

debug_log_error: Método utilizado para exibir erros do sistema.

initialize: Inicia os componentes do Allegro.

init: Inicia as variáveis globais do jogo.

car_size: Retorna o tamanho do carro de acordo com sua distância do jogador (0 longe, 1 perto).

draw_scene: Desenha os elementos da cena, juntamente com o jogador e seus respectivos oponentes.

quit_application: Destrói os componentes do Allegro e desaloca a memória usada por eles.

handle_input: Interpreta os comandos do usuário lidos pelo teclado.

AABB_collision_test: Axis-Aligned Bounding Box é o método mais simples para detecção de colisões em duas dimensões (para objetos alinhados em um mesmo eixo).

tick: A função “tick” é executada uma vez por frame, ou seja, caso o jogo esteja rodando a 100 FPS (frames per second) a função será executada 100 por segundo.

main: A principal função de qualquer programa em C, o ponto de partida. A função “main” é responsável por inicializar os componentes do Allegro e a cada frame, checar os comandos do usuário e redesenhar o cenário com as propriedades alteradas.

Além das rotinas do programa, temos as funções da biblioteca de cor, utilizada no jogo para simular a transição de dia e noite.

new_color: Cria uma nova cor com os componentes passados por parâmetro.

day_sky_color: “Constante” que representa a cor do céu durante o dia.

night_sky_color: “Constante” que representa a cor do céu durante a noite.

clamp: Impede que o valor “v” seja maior que o “max” e menor que o “min”.

clamp01: Impede que o valor “v” seja maior que 1 e menor que 0.

normalize: Divide cada componente da cor por 255, afim de obter um valor correspondente entre 0 e 1.

mul: Multiplica 2 cores de forma vetorial.

add: Soma 2 cores

subtract: Subtrai “b” de “a”.

color_2_allegro_color: Converte uma “Color” para ALLEGRO_COLOR.

lerp: Interpola entre “a” e “b” de acordo com “t”.

lerp_color: Interpola entre as cores “a” e “b”.

O jogo foi desenvolvido na forma de módulos, cada módulo com sua exclusiva função, por exemplo, não há nenhuma função relacionada a renderização no método **handle_input**. A vantagem dessa técnica é a fácil manutenção do programa, visto que cada rotina é responsável apenas por uma parte do jogo.